UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1241

Programme 5
Automatique, Productique,
Traitement du Signal et des Données

FAST SOLUTION OF THE OBSTACLE PROBLEM

Mabel A. MEDINA
Roberto L. V. GONZALEZ

Juin 1990

* R R - 1 2 4 1 *

FAST SOLUTION OF THE OBSTACLE PROBLEM

SOLUTION RAPIDE DU PROBLEME DE L'OBSTACLE

Mabel A. Medina * - Roberto L. V. González *+

---

* Instituto de Matemática Beppo Levi, Facultad de Cs. Exactas, Ingenieria y Agrimensura, Universidad Nacional de Rosario, Avda. Pellegrini 250, 2000 Rosario - Argentine.

+ This paper is included in the activities developed in the frame of the Cooperation Project INRIA - Instituto de Matemática Beppo Levi (coordinators of the project : E. Rofman - R. González).

ABSTRACT

The purpose of this paper is to present a fast algorithm for the numerical solution of the one-dimensional obstacle problem. It is proven that it converges in a finite number of steps and application examples that show its efficiency are presented.

RESUME

Nous présentons ici un algorithme rapide pour la solution numérique du problème de l'obstacle. On démontre qu'il converge après un nombre fini de pas et on présente quelques exemples d'application où la performance de l'algorithme est évidente.

CONTENTS

1. DESCRIPTION OF THE CONTINUOUS PROBLEM

Let be the interval $I = [0,L]$ and a function $\psi : [0,L] \to \Re$, $\psi \in W = H^2(0,L)$.
The problem of finding the upper envelope of function ψ (the minimum concave function that satisfies $y(s) \geq \psi(s)$ $\forall$ $s \in [0,L]$) can be reduced to finding the solution to the variational inequality:

$$\begin{aligned} & y \geq \psi \\ & (y', v' - y') \geq 0 \qquad \forall v \in W \end{aligned} \tag{1}$$

where (u,v) denotes the scalar in $L^2(0,L)$ of two elements u,v

This solution is characterized by the conditions:

$$y \geq \psi \tag{2}$$

$$y'' \leq 0 \quad \text{en } L^2(0,L) \tag{3}$$

$$y(x) > \psi(x) \Rightarrow \exists\, \delta > 0 \ / \ y''(t) = 0 \quad \forall\, t \in (x-\delta, x+\delta). \tag{4}$$

For these results see [1]

2. DISCRETIZATION OF THE CONTINUOUS PROBLEM

For the numerical solution of problem (1), this variational inequality is discretized by using the finite differences method.

2.1 Description of the Discretization Procedure

2.1.1 *Interval I is replaced by the set* I_h

$$I_h = \left\{ x_i \ / \ x_i = (i-1)\,h \ , \quad i = 1 \ , N+1 \ ; \qquad h = \frac{1}{N} \right\}$$

2.1.2 *Space W is replaced by space W_h*

$$W_h = \{ y() \ / \ y : I_h \to \Re^{N+1} \}$$

An element of W_h will then be identified by a vector "y" of components

$$y_i, \quad i = 1, N+1.$$

2.1.3 *Discretization of differential inequalities*

★ Condition $y'' \leq 0$ is discretized in the following way:

$$y_{i-1} - 2\, y_i + y_{i+1} \leq 0 \ , \qquad i = 2, \ldots, N \tag{5}$$

★ Condition $y \geq \psi$ is trivially discretized:

$$y_i \geq \psi(x_i) \ , \qquad i = 1, \ldots, N+1 \tag{6}$$

2.2 Presentation of the discretized problem

Three equivalent presentations of the discretized problem will be given here:

P_1^h : Find the minimum element of W_h that verifies (5) and (6).

P_2^h : Find the only element that verifies (5), (6) and (7), where:

$$\left.\begin{array}{l} y_1 = \psi(x_1) \quad , \qquad y_{N+1} = \psi(x_{N+1}) \\ \\ y_i > \psi(x_i) \Rightarrow y_{i-1} - 2\, y_i + y_{i+1} = 0 \end{array}\right\} \tag{7}$$

P_3^h Find the unique fixed point of the operator:

$$A_h : W_h \to W_h$$

where:

$$
\begin{cases}
(A_h y)_i = \psi(x_i) & \text{if} \quad i = 1 \quad \text{or} \quad i = N+1 \\
(A_h y)_i = \max\left(\psi(x_i)\,,\ \frac{1}{2}(y_{i-1} + y_{i+1})\right) & \text{for} \quad i = 2, \ldots, N
\end{cases}
\tag{8}
$$

2.3 Existence and uniqueness of solution to the discretized problem

The above presented problem are equivalent in the sense that they have a unique solution but each of them has a particular advantage. P_1^h is the natural way to obtain the discrete problem from the continuous one; P_2^h is very useful to prove the convergence of discrete solution to the original solution and P_3^h is the most convenient set-up to see the discrete problem from the algorithmical point of view.

Theorem 1. *Problems P_1^h, P_2^h, and P_3^h are equivalent and have a unique solution (the discretized problem will be called P^h and its solution $\bar{y}_h$).*

2.4 Iterative Computation of the Solution

2.4.1 *Preliminary definitions*

We say that y is a supersolution (subsolution) of problem P^h iff

$$
y_i \geq (A_h y)_i \qquad (y_i \leq (A_h y)_i)
\tag{9}
$$

Theorem 2. *Operator A_h is monotone, i.e.*

$$
y \geq \hat{y} \;\Rightarrow\; (A_h y) \geq A_h \hat{y}
\tag{10}
$$

and contractive in the following sense:

$$\left.\begin{array}{l} \| A_h y - A_h \tilde{y} \| \le \| y - \tilde{y} \| \\ \| A_h^p y - A_h^p \tilde{y} \| \le \beta \| y - \tilde{y} \| \end{array}\right\} \tag{11}$$

$$\text{with } \beta < 1 , \quad \| y \| = \max_{i = 1,N+1} | y_i | , \quad p=[\tfrac{N}{2}] \tag{12}$$

Based on the presentation P_3^h of the discretized problem and on the properties stated above for the operator A_h, the following recurrence for the computation of $\overline{y}_h$ is defined:

Algorithm A0

Step 0: set $y_j^0 = \hat{y}_j \quad \forall j = 1,N+1 , \quad \hat{y} \in W_h$ and set $m=0$

Step 1: set $\quad y^{m+1} = A_h y^m$ (13)

Step 2: set m=m+1 and go to step 1

With respect to this algorithm, the following result holds:

Theorem 3. *The sequence y^m converges to the unique solution $\overline{y}$ of P^h from any initial value $\hat{y} \in W_h$; if $\hat{y}$ is a supersolution (subsolution), sequence y^m is decreasing (increasing). The following error bound is also valid:*

$$\| \overline{y} - y^m \| \le \beta^{[\frac{m}{p}]} \| \overline{y} - y^0 \| \tag{14}$$

Remark 2.1 Although iteration (13) is an admissible computation algorithm, its convergence can be extremely slow if N is very large.

Remark 2.2: The proofs of Theorems 1, 2 and 3 are outside the main stream of this paper and are left as an exercise to the reader (the general lines of the proof can be seen in [3])

3. FAST COMPUTATIONAL ALGORITHM

We will propose a fast algorithm that is inspired on the property of the function which is solution to the original problem of being linear in the zone where $y(x) > \psi(x)$.

3.1 Preliminary definitions

- $\forall\ y \in W_h$ we define $\gamma(y)$

$$\gamma(y) = \{\ i\ /\ y_i = \psi(x_i)\ \} \tag{15}$$

Let $S \subset \{\ 1,2, \ldots, N+1\ \}$

- $i^-(S) = \max\ \{\ j\ /\ j < i\ ,\ \ j \in\ S\ \}$ (16)

- $i^+(S) = \min\ \{\ j\ /\ j > i\ ,\ \ j \in\ S\ \}$ (17)

- $F(y,S,i) = \Big(\ (i^+ - i)\ y_{i^-} + (i\ - i^-) y_{i^+}\ \Big)\ /\ (i^+ - i^-)$ (18)

- $M : W_h \times S \rightarrow \{0,1\}^{N-1}$

$$\Big(M(y,S)\Big)_i \begin{cases} = 0 & \text{if } y_i \geq F(y,S,i) \\ = 1 & \text{if } y_i < F(y,S,i) \end{cases} \tag{19}$$

Remark 3.1: Instead of using the expression (5) to compute the discrete version of the second derivative of y, we use (due to the fact that y_h is linear between i^+ and i^-) the equivalent relation:

$$y_i \geq F(y,S,i)$$

in place of (5)

3.2 Description of the fast algorithm

Algorithm A1

Step 0: set $y_j^0 = \psi(x_j)$ $\quad \forall j = 1,N+1$

Step 1: set $\nu = 0$, $i = 2$, $S^0 = \{1,2, \ldots , N+1\}$

Step 2: if $M(y^\nu, S^\nu)_i = 1$, then :

$$i^{\nu+1} = i \tag{20}$$

$$S^{\nu+1} = \gamma(y^\nu) \;/\; i^{\nu+1} \tag{21}$$

$$y_j^{\nu+1} = \psi(x_j) \quad \forall j \in S^{\nu+1} \tag{22}$$

$$\begin{aligned} y_j^{\nu+1} &= F(y^\nu, \; S^{\nu+1}, \; j) = \\ &= F(y^{\nu+1}, S^{\nu+1}, j) \quad \forall j \notin S^{\nu+1} \end{aligned} \tag{23}$$

and go to step 3

if $M(y^\nu, S^\nu)_i = 0$,

if $i^+(S^\nu) = N+1$

then $\overline{\nu} = \nu$ and stop ($y^{\overline{\nu}} = \overline{y}_h$).

if $i^+(S^\nu) < N+1$,

then $i = i^+(S^\nu)$ and go to step 2.

Step 3. Set $\nu = \nu+1$. if $i^-(S^\nu) = 1$ and $i^+(S^\nu) \leq N$, set $i = i^+(S^\nu)$

if $i^-(S^\nu) \geq 2$, set $i = i^-(S^\nu)$

if $i^-(S^\nu) = 1$ and $i^+(S^\nu) = N+1$,

set $\overline{\nu} = \nu$ and stop ($y^{\overline{\nu}} = \overline{y}_h$)

Step 4 : set $l(\nu) = i-1$ and go to step 2

3.3 Convergence of Algorithm A1

To prove the convergence, we will show that **algorithm A1 generates an increasing** sequence of functions which in a finite number of **steps converges to the solution** which satisfies (5), (6) and (7).

Lemma 1 : $$y_j^\nu \geq \psi(x_j) \quad \forall j = 1 , N+1 \quad \forall \nu = 0,1, \ldots . \tag{24}$$

Proof : Obvious for $\nu = 0$. For each $\nu \geq 1$, vector y^ν **is defined when the test** of step 2 is positively verified, consequently:

$$y_j^\nu = \psi(x_j) \quad \forall j \in S^\nu \tag{25}$$

Also, if $j \notin S^\nu$, two options can arise:

a) $j = i^\nu$, which implies $(My^{\nu-1}, S^{\nu-1})_{i^\nu} = 1 \Rightarrow y_{i^\nu}^{\nu-1} < F(y^{\nu-1}, S^{\nu-1}, i^\nu)$

but $\psi(x_{i^\nu}) = y_{i^\nu}^{\nu-1}$, then

$$\psi(x_{i^\nu}) = y_{i^\nu}^{\nu-1} < F(y^{\nu-1}, S^{\nu-1}, i^\nu) = F(y^{\nu-1}, S^\nu, i^\nu) = F(y^\nu, S^\nu, i^\nu) = y_{i^\nu}^\nu \tag{26}$$

b) $j \neq i^\nu$, which implies $j \notin S^{\nu-1}$, consequently :

$$y_j^\nu = F(y^{\nu-1}, S^\nu, j) = F(y^\nu, S^\nu, j)$$

it is clear that y_j^ν, being between $j^-(S^\nu)$ and $j^+(S^\nu)$ **the linear interpolation** of the values:

$$y^\nu_{j^+(S^\nu)} \;,\; y^\nu_{j^-(S^\nu)}$$

is also the linear interpolation of the values:

$$y^\nu_{j^+(S^{\nu-1})} \;,\; y^\nu_{j^-(S^{\nu-1})}$$

because

$$j^-(S^{\nu-1}) \geq j^-(S^\nu)$$

$$j^{+}(S^{\nu-1}) \leq j^{+}(S^{\nu})$$

then

$$y_j^{\nu} = F(y^{\nu}, S^{\nu-1}, j)$$

but since $S^{\nu-1} = S^{\nu} \cup \{ i^{\nu} \}$, there are two options for $j^{\pm}(S^{\nu-1})$:

b$_1$) $j^{\pm}(S^{\nu-1}) \in S^{\nu}$ in which case $y^{\nu}_{j^{\pm}(S^{\nu-1})} = \psi(x_{j^{\pm}(S^{\nu-1})}) = y^{\nu-1}_{j^{\pm}(S^{\nu-1})}$

or

b$_2$) $j^{\pm}(S^{\nu-1}) = i^{\nu}$ in which case $y^{\nu-1}_{j^{\pm}(S^{\nu-1})} = \psi(x_{j^{\pm}(S^{\nu-1})}) < y^{\nu}_{j^{\pm}(S^{\nu-1})}$

by virtue of (26).

Then, as y_j^{ν} is the interpolation of the values $y^{\nu}_{j^{+}(S^{\nu-1})}$, $y^{\nu}_{j^{-}(S^{\nu-1})}$

$$y_j^{\nu} = F(y^{\nu}, S^{\nu-1}, j) \geq F(y^{\nu-1}, S^{\nu-1}, j) = y_j^{\nu-1} \tag{27}$$

by induction, $y_j^{\nu-1} \geq \psi(x_j)$, with which (24) is proved.

□

Corollary: $$y_j^{\nu} \geq y_j^{\nu-1} \qquad \forall j\ ,\ \forall \nu = 1, \ldots$$

Proof: Evident by virtue of (25)-(27).

□

Lemma 2: $\forall \nu \geq 1$, if $l(\nu) \geq 2$ then $\forall\, a\ /\ 2 \leq a \leq l(\nu)$

$$y_a^{\nu} \geq F(y^{\nu}, S^{\nu}, a) \tag{28}$$

Proof. We will do it by induction :

a) for $\nu = 1$, vector y^{ν} verifies:

$$y_j^{\nu} = y_j^{0} = \psi(x_j) \qquad \forall j \neq i^{\nu}$$

since i^{ν} is the index where the test in step 2 is satisfied; it also satisfies :

$$M(S^0, y^0)_j = 0 \qquad \forall\ 2 \leq j \leq i^{\nu}-1$$

in particular $\quad \forall\ 2 \leq j \leq (i^1)^-(S^1)-1 = i^{\nu}-2;$

Since $y^1_j = y^0_j \quad \forall\ 1 \leq j \leq i^1-1$

and also

$$j^+(S^1) = j+1 = j^+(S^0) \quad \forall\ j \leq i-2$$

$$j^-(S^1) = j-1 = j^-(S^0) \quad \forall\ j \leq i-2$$

we have

$$F(y^0, S^0, j) = F(y^1, S^1, j) \qquad \forall\ 2 \leq j \leq i-2$$

but

$$M(S^0, y^0)_j = 0 \qquad \Rightarrow \qquad y^0_j \geq F(y^0, S^0, j)$$

from where

$$y^1_j = y^0_j \geq F(y^0, S^0, j) = F(y^1, S^1, j) \qquad \forall\ 2 \leq j \leq (i^1)^-(S^1)-1 = I(1)$$

and then (28) is valid.

b) by induction, we will assume that (28) is valid for ν.

Let be $\nu+1$ with $I(\nu+1) \geq 2$.

The index $i^{\nu+1}$ is defined when the test in step 2 is positively verified; we divide the analysis in considering the possible cases for $i^{\nu+1}$.

$$i^{\nu+1} < i^{\nu} \qquad \text{or} \qquad i^{\nu+1} > i^{\nu}$$

b_1) $i^{\nu+1} < i^{\nu}$

From the special structure of the algorithm, where in loop 2-3-4 the analysed index i is increasing and only takes a lower value than i^{ν} when the loop is started, it must be:

$$i^{\nu+1} = (i^{\nu})^{-}(S^{\nu})$$

By the definition of $y^{\nu+1}$, we have that

$$y_j^{\nu+1} = F(y^{\nu+1}, S^{\nu+1}, j) \quad \forall\ j \notin S^{\nu+1}$$

so in order to verify the validity of (28) it should be enough the analysis of the cases where $j \in S^{\nu+1}$

Let $2 \leq j \in S^{\nu+1}$ such that

$$j \leq (i^{\nu+1})^{-}(S^{\nu+1}) - 1 = l(\nu+1)$$

By the definition of $y^{\nu+1}$, we have that

$$y_p^{\nu} = y_p^{\nu+1} \qquad \forall\ 1 \leq p \leq (i^{\nu+1})^{-}(S^{\nu+1}) \tag{29}$$

and also

$$p^{\pm}(S^{\nu+1}) = p^{\pm}(S^{j}) \qquad \forall\ 2 \leq p \leq (i^{\nu+1})^{-}(S^{\nu+1}) - 1$$

therefore

$$F(y^{\nu+1}, S^{\nu+1}, j) = F(y^{\nu}, S^{\nu}, j)$$

and in consequence, by virtue of (29) and the induction hypothesis, it holds

$$y_j^{\nu+1} = y_j^{\nu} \geq F(y^{\nu}, S^{\nu}, j) = F(y^{\nu+1}, S^{\nu+1}, j)$$

that is what we wanted to prove.

$b_2)$ $i^{\nu+1} > i^{\nu}$

Similarly to the previous reasoning, it is necessary to analyse only the case

$$2 \leq j,\ j \in S^{\nu+1},\ j \leq (i^{\nu+1})^{-}(S^{\nu+1}) - 1$$

For these values of j we have

$$j^{\pm}(S^{\nu+1}) = j^{\pm}(S^{\nu}) \tag{30}$$

at the same time, it always holds that

$$y_p^{\nu} = y_p^{\nu+1} \qquad \forall\ 1 \le p \le (i^{\nu+1})^{-}(S^{\nu+1}) \tag{31}$$

We divide the analysis in two cases

* if $j \le l(\nu) = (i^{\nu})^{-}(S^{\nu})-1$

$y_j^{\nu} \ge F(y^{\nu}, S^{\nu}, j)$ is satisfied by induction, so that

$y_j^{\nu+1} = y_j \ge F(y^{\nu}, S^{\nu}, j) = F(y^{\nu+1}, S^{\nu+1}, j)$ by virtue of (30) and (31).

* if $l(\nu) < j \le (i^{\nu+1})^{-}(S^{\nu+1})-1$

$y_j^{\nu} \ge F(y^{\nu}, S^{\nu}, j)$ because the test in step 2 was not satisfied, so that by virtue of (30) and (31),

$$y_j^{\nu+1} = y_j^{\nu} \ge F(y^{\nu}, S^{\nu}, j) = F(y^{\nu+1}, S^{\nu+1}, j)$$

is satisfied, and then the proof by induction is completed.

□

Theorem 4. *Algorithm A1 converges in a finite number of steps to the solution* $\bar{y}_h$ *of problem* P^h

Proof. Since each time a new index ν is generated, the set S^{ν} is reduced by an element, it is obvious that the algorithm must end in a finite number of steps.
Let $\bar{\nu}$ be the value of the index corresponding to the last time a point of set S^{ν} is removed.
By lemma 1,

$$y_j^{\bar{\nu}} \ge \psi(x_j) \qquad \forall\ j = 1, N+1$$

is satisfied; moreover, by the construction of element $y^{\overline{\nu}}$, we have:

$$y_j^{\overline{\nu}} > \psi(x_j) \quad (\text{ i.e. } j \notin S^{\nu}) \quad \Rightarrow \quad y_j^{\overline{\nu}} = F(y^{\overline{\nu}}, S^{\overline{\nu}}, j)$$

It remains only to prove that $y^{\overline{\nu}}$ satisfies

$$y_j^{\overline{\nu}} \geq F(y^{\overline{\nu}}, S^{\overline{\nu}}, j) \qquad \forall j \in S^{\overline{\nu}}$$

By lemma 2, we know that

$$y_j^{\overline{\nu}} \geq F(y^{\overline{\nu}}, S^{\overline{\nu}}, j) \qquad \forall j \in S^{\overline{\nu}}, \quad \forall\ 2 \leq j \leq \left((i^{\overline{\nu}})^{-}-1\right)$$

By entering the algorithm again in step 2, the points $p \in S^{\overline{\nu}}$ such that $p \geq (i^{\overline{\nu}})^{-}-1$ are analysed; so, as $\overline{\nu}$ is the last index, this implies that for those points it is always verified that :

$$\left(M(y^{\overline{\nu}}, S^{\overline{\nu}})\right)_j = 0$$

that is

$$y_j^{\overline{\nu}} \geq F(y^{\overline{\nu}}, S^{\overline{\nu}}, j)$$

as a consequence $y^{\overline{\nu}}$ verifies (5) and then it coincides with the unique solution of P^{n}. In this way we have proved that the algorithm finishes in a finite number of steps.

$\square$

4. PRACTICAL IMPLEMENTATION OF THE ALGORITHM

The description of algorithm A1 is a detailed theoretical version that allows us to prove its convergence. By observing that actually the algorithm operates on values y_j^{ν} / $j \in S^{\nu}$, it is obvious that it is only necessary to preserve those values (which constitute a set of decreasing cardinality), because between them, vector y^{ν} has a linear behaviour. In this way, the algorithm can take the following practical implementation:

Algorithm A1':

Step 0: set $NT = N+1$

$x_j = (j-1)h\ , \quad \forall\ j=1, NT$

$y_j = \psi(x_j)\ , \quad \forall\ j=1, NT$

$i=2$

Step 1: If $i+1 > NT$, stop (the solution $\overline{y}_h$ is the linear interpolation of values y_j on the points x_j, $j=1, NT$)

If $i < NT-1$ go to step 2.

Step 2: set $D_2 = (y_{i+1} - y_i)/(x_{i+1} - x_i)$

$D_1 = (y_i - y_{i-1})/(x_i - x_{i-1})$

if $D_2 \leq D_1$, set $i = i+1$, and go to step 1.

if $D_2 > D_1$, set $x_j = x_{j+1} \quad \forall\ j = i, \ldots, NT-1$.

$y_j = y_{j+1} \quad \forall\ j = i, \ldots, NT-1$.

set $NT = NT-1$

$i = \max(2, i-1)$, and go to step 1.

5. APPLICATION

5.1 Examples

We present here some examples corresponding to different functions ψ and various sizes of N.

Example 1:

The function ψ has the following form

$$\psi = - \text{sen}\,(\pi\, t)\ \ \text{sen}\,(\pi\, q\, t) \qquad\qquad t \in [0,1]$$

the problem was solved for the values of q=19, N=1000. See Figure 1.

Example 2:

The function ψ has a linear behavior between values generated in a random fashion. The results are shown in Figure 2 for the value N=200.

Example 3:

It corresponds to a practical problem originated in the analysis of the solidification of steel in a continuous casting machine. Results are shown in Figure 3.

5.2 Time comparison between Algorithms A1 and A0

In the following tables are shown the times of computation employed by Algorithm A1 and A0 (see also Figures 4-6). Algorithm A0 is stopped when the relative error with respect to solution $\bar{y}_h$ is smaller than ε. The values corresponds to different parameters N and ε.

It can be observed that times for Algorithm A1 are independent of ε, and are asymptotically a linear function of N. Also, it is evident that Algorithm A1 improves dramatically the performance of the computational procedure to solve the obstacle problem. For $\varepsilon = 10^{-5}$ and N= 400 the improvement is of 99.95 %.

TABLE 1

	Times							
ε	10^{-2}		10^{-3}		10^{-4}		10^{-6}	
N	A1	A0	A1	A0	A1	A0	A1	A0
20	5″	5.6″	5″	6.29″	5″	7.03″	5″	8.51″
40	5.2″	8.07″	5.2″	9.88″	5.2″	11.69″	5.2″	15.43″
100	5.71″	49.81″	5.71″	73.40″	5.71″	1′ 37″	5.71″	2′ 25″
200	6.86″	4′ 33″	6.86″	7′55″	6.86″	11′17″	6.86″	18′1″
400	8.51″	90′17″	8.51″	2^h15′	8.51″	3^h1′	8.51″	4^h30′

TABLE 2

N	Times (A1)
100	5.71″
200	6.86″
400	8.51″
1000	13.73″
4000	39.76″
8000	74.42″
10000	91.78″

Remark 5.1: These results were obtained in a computer PC-IBM/AT, and the programs were written in Fortran77.

6. CONCLUSIONS

We have presented here a fast algorithm for the numerical solution of the one-dimensional obstacle problem. We have proved that it converges in a finite number of steps. Examples of comparison show its efficiency and usefulness.

The methodology above presented can be extended to deal with a problem issued from a field of image processing, that is the problem of fiding the least convex envolope of a closed curve. This will be the subject of a paper in preparation.

APPENDIX

Figures

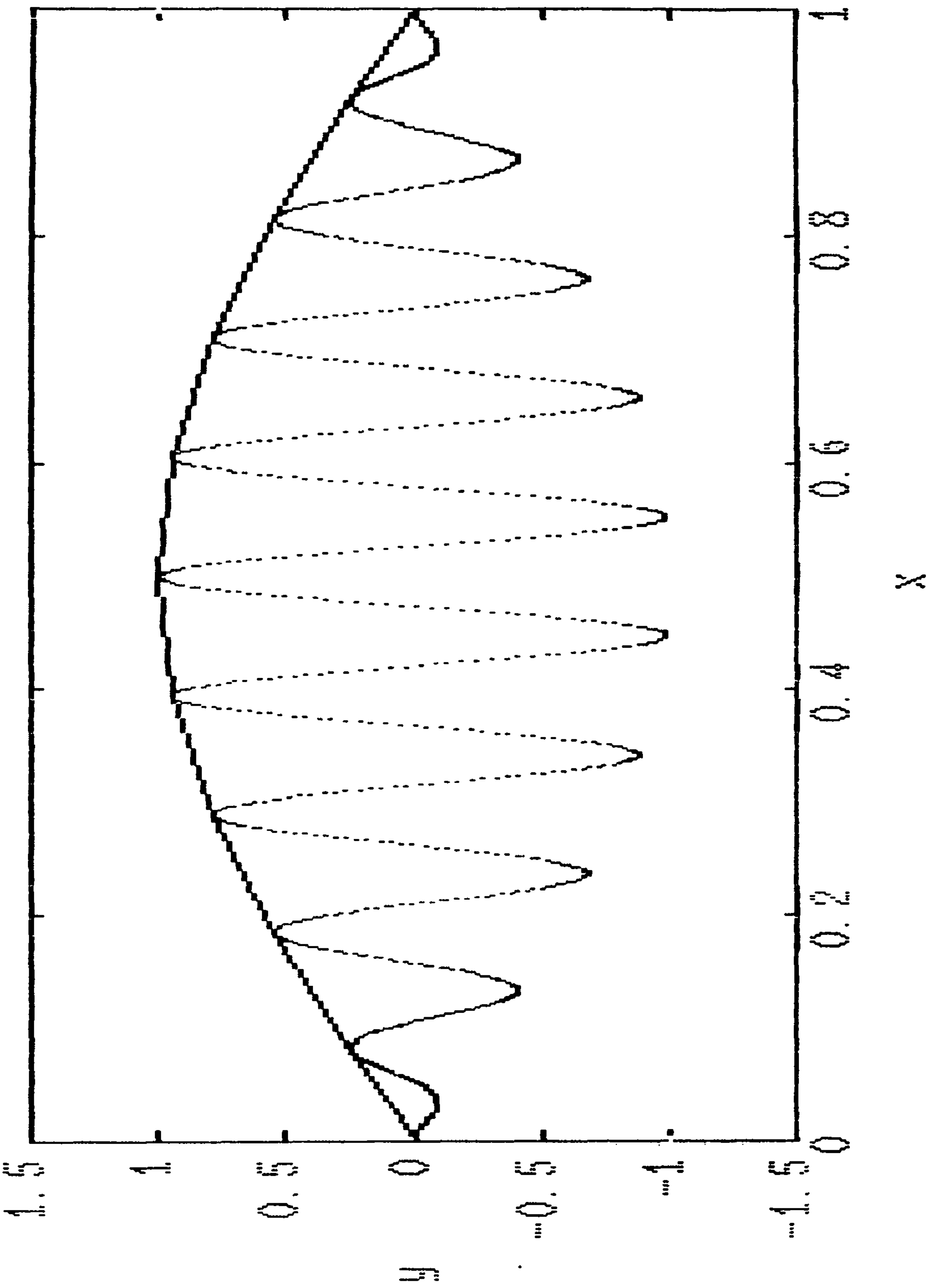

Figure 1

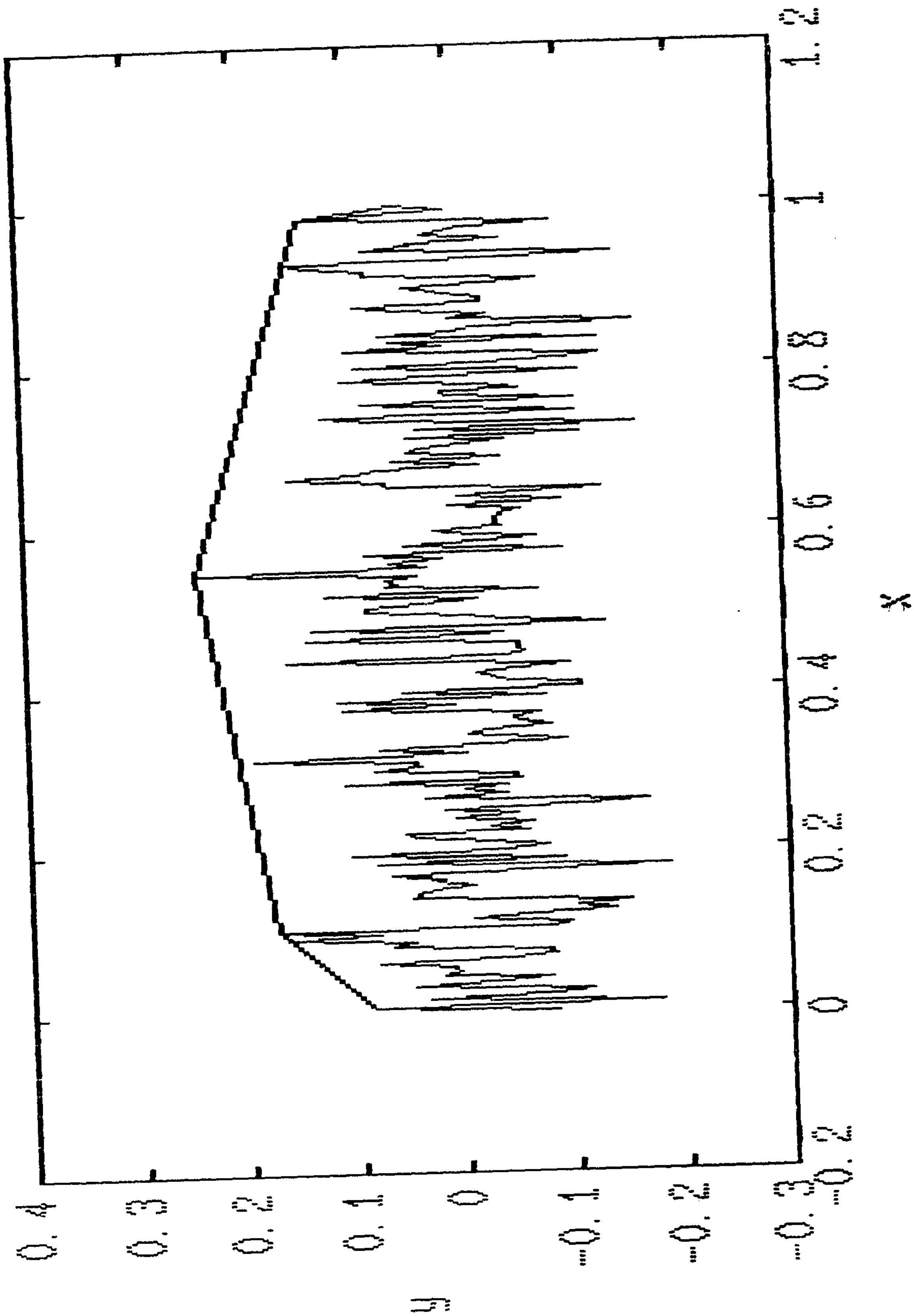

Figure 2

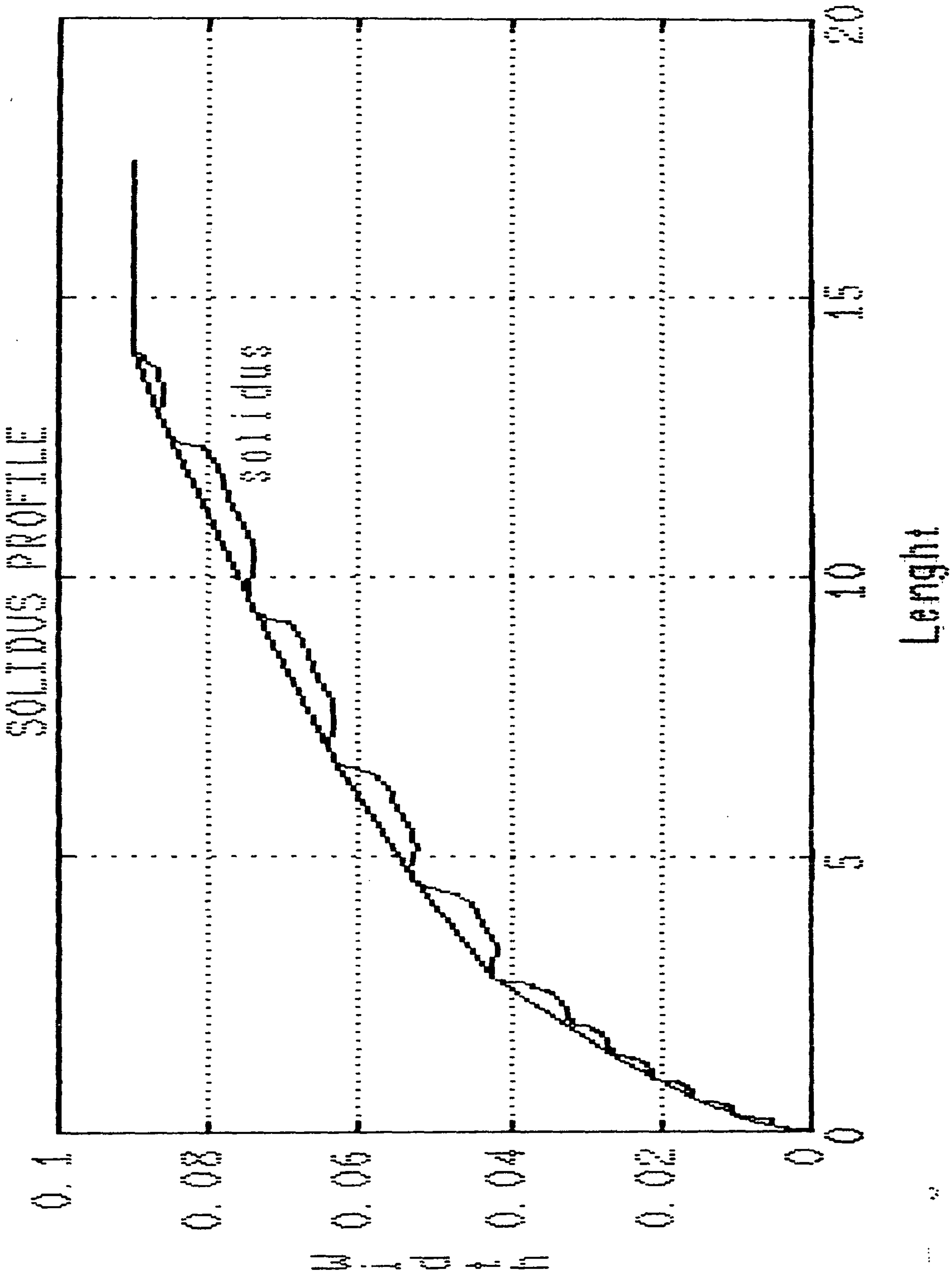
SOLIDUS PROFILE
solidus
Lenght
Width
0
0.02
0.04
0.06
0.08
0.1
0
5
10
15
20

Figure 3

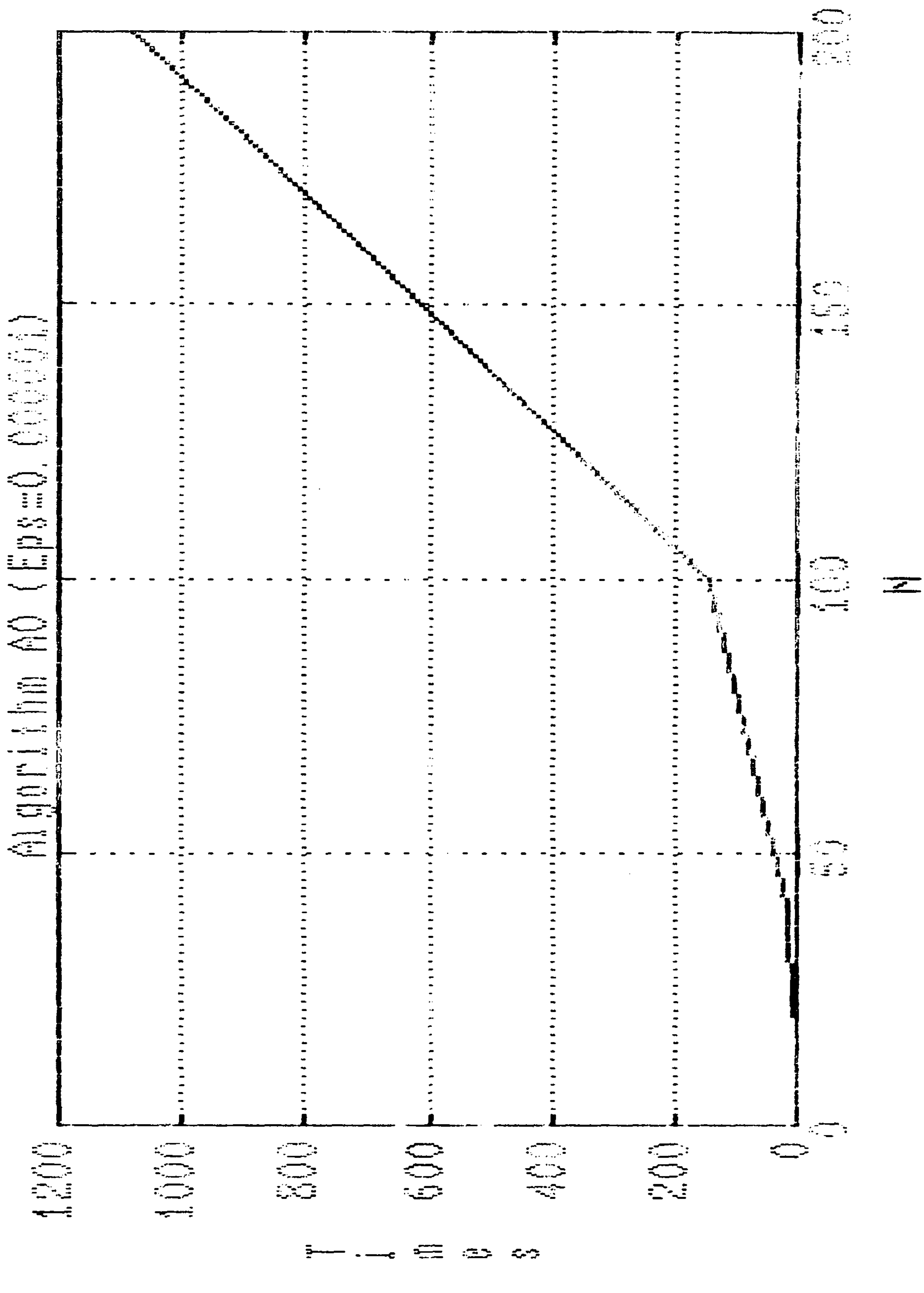
Algorithm A0 (Eps=0.00001)
Time
N
0
200
400
600
800
1000
1200
0
50
100
150
200

Figure 4

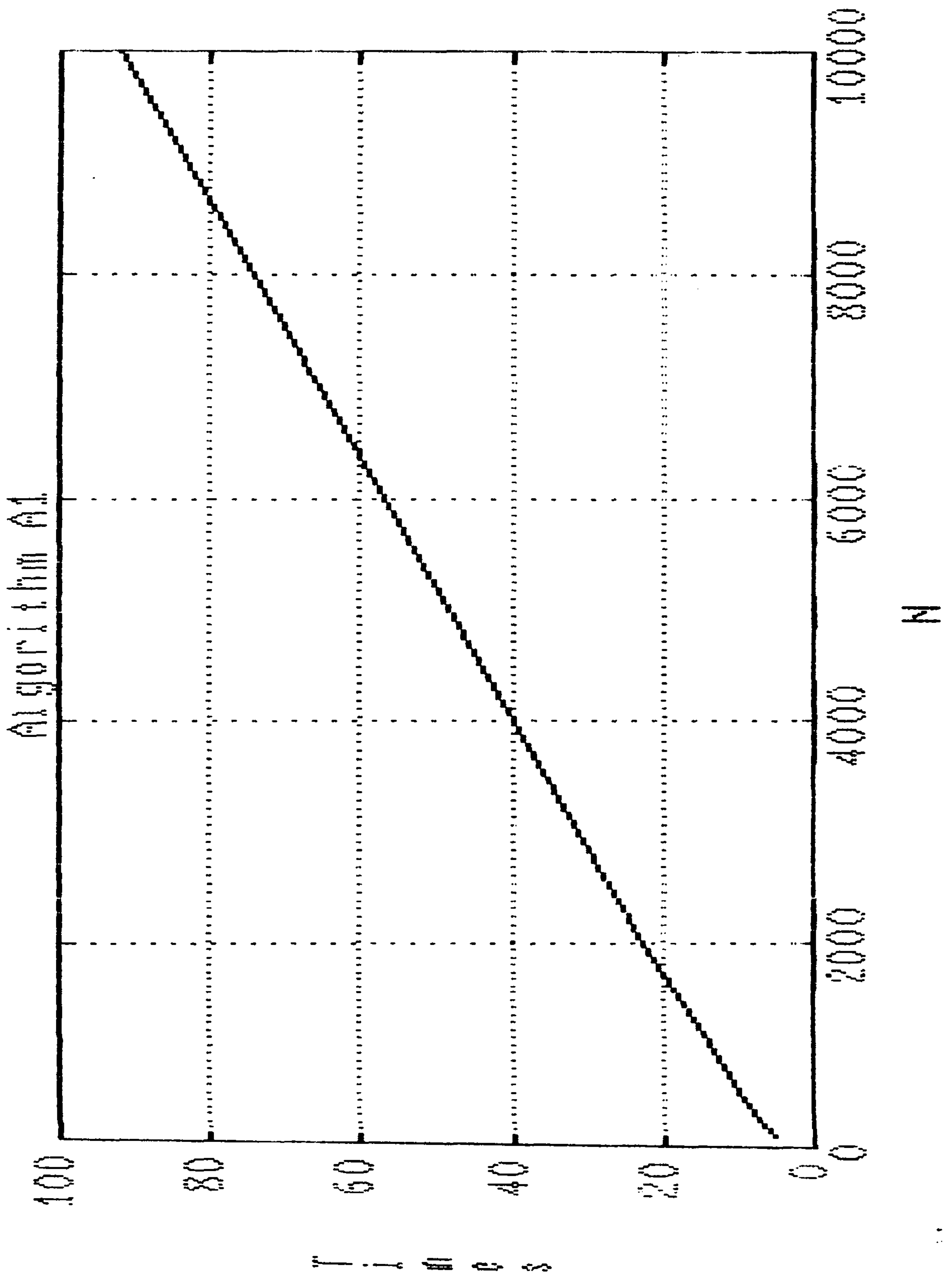

Figure 5

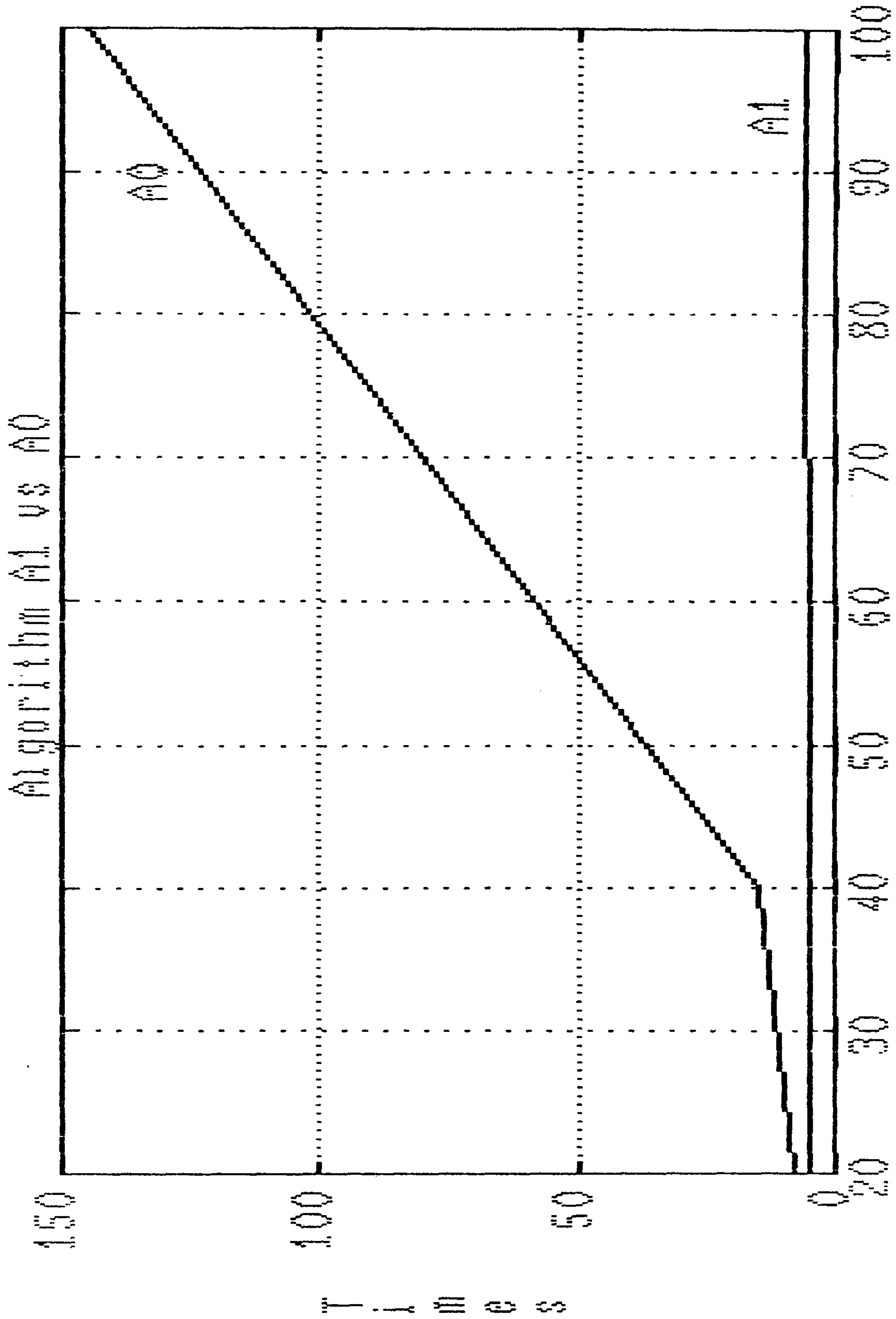
Algorithm A1 vs A0
A0
A1
150
100
50
0
20
30
40
50
60
70
80
90
100
N
Times

Figure 6

REFERENCES

[1] Glowinski R. *Numerical Methods for Nonlinear Variational Problems* (Springer-Verlag, New York, 1984)

[2] Glowinski R., Lions J. L., Tremolières R: *Numerical Analysis of Variational Inequalities* (North Holland, Amsterdam, 1981).

[3] González R., Rofman E.: On deterministic control problems: an approximation procedure for the optimal cost. Part I and II. *SIAM Journal on Control and Optimization*, 23, pp. 242-285, 1985.

[4] Mosco U. An introduction to the approximate solution of variational inequalities. In *Constructive Aspects of Functional Analysis*, Vol II, pp. 497-682 (Edizioni Cremonese, Roma, 1971).

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399